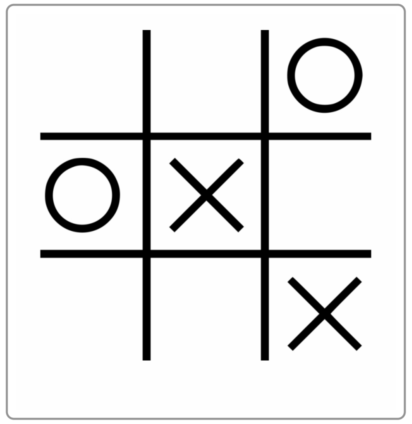
Assets

Tic tac toe (2020)

Deadline: 30 Oct 23:59 **Bonus**

The goal of this exercise to write a computer program that trains two players to play tic-tac-toe by playing against each other.

To obtain credits for this exercise, upload a PDF-document presenting your results (details below). Use the upload button at the top of this page. Merge the PDF-files you have uploaded for Homework 3 into one document. Make a front page, and attach the computer code you have used in appendices. Submit your merged PDF-document to URKUND, before the deadline.

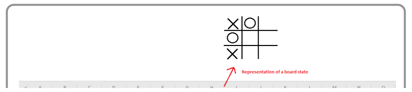
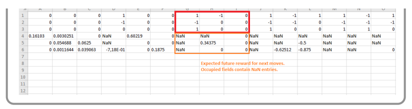


Two players learn how to play tic tac toe with Q-learning. Train the two players simultaneously, playing against each other. The goal is to find a Q-table ensuring that the player never loses, see Section 11.5 in the Lecture notes. Use the ε-greedy policy, start with a suitable value of ε , and let ε tend to zero as training progresses.

Each player has his/her own Q-table. For an efficient implementation, do not initialise your Q-tables for every possible state of the tic tac toe board. Instead only initialise Q-entries when they are enountered for the first time during training. To this end, write a function that checks whether a board configuration occurred previously, or not. If not, add it to your Q-table and initialise it.

You need to upload two csv files. One should be named "player1.csv" and describe the Q-table of the player going first. The other one should be called "player2.csv" for the player going second. The csv files should have the following structure: The first three lines should consist of all the board states known to your Q-table, horizontally appended. A board position is implented as a 3-by-3 block, where an "X" is represented by a "1", an "O" by a "-1" and an empty field by a "0". The next three lines mimic the structure of the boards in the first three lines, except here an entry represents the expected value for a move in that location. Move values for occupied locations should be represented by NaN entries.

The following image visualises the structure on the file "examplecsv.csv", which is also available to

download  Note that the specific numerical values in bottom three lines download. 

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	0	0	0	1	0	0	1	-1	0	1	-1	1	1	-1	1
2	0	0	0	-1	0	0	-1	0	0	-1	-1	0	-1	-1	-1
3	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0
4	0.16103	0.0030251	0	NaN	0.60219	0	NaN	NaN	0	NaN	NaN	NaN	NaN	NaN	NaN
5	0	0.054688	0.0625	NaN	0	0	NaN	0.34375	0	NaN	NaN	-0.5	NaN	NaN	NaN
6	0	0.0011644	0.039063	-7,18E-01	0	0.1875	NaN	0	0	NaN	-0.62512	-0.875	NaN	NaN	0

Expected future reward for next moves.
Occupied fields contain NaN entries.

Note that the specific numerical values in bottom three lines are not meaningful, they are only meant to demonstrate the structure of the file.

Hints: To easily obtain the desired .csv structure in Matlab, maintain the Q-table as a cell array with 2 rows and as many columns as there are known board states. In the first row, save a board state in every cell as a 3x3 matrix and in the second row the corresponding expected future rewards also as a 3x3 matrix with NaN entries on occupied fields. You can then use the function cell2mat followed by csvwrite to generate a .csv file with the right structure. The matlab function isnan can be very useful when debugging errors with the NaN entries. Note that Matlab evaluates arithmetic operations where one component is NaN entirely as NaN.

When your CSV-files are in place, click the button to submit your Q-matrices.
[0 attempts]

(not graded)

Presentation of results

Upload a one page PDF-document presenting and discussing your results. Illustrate how training progresses by plotting three curves: the fraction of games player x wins, the fraction of games player o wins, and the fraction of games that end in a draw, all as a function of training iterations.

Submit at most one A4 page with 12pt single-spaced text, and with 2cm margins. Each page may contain at most one Figure and/or one Table with the corresponding Figure and/or Table caption, in addition to the text discussing the results shown in the Figure/Table. It is not necessary to write a full page for each of the four problems in homework 3, but you must explain/describe what you have done and clearly state your answers/results to the questions, as well as your conclusions. When necessary you must discuss possible errors and inaccuracies in your results. Plots/graphs must have legible axis labels and tic labels. All symbols and lines must be explained in the Figure or in a caption. The Figure may consist of separate panels, label them 'a', 'b', and so forth.